The `fontloader-luaotfload` package

Ulrike Fischer*

August 20, 2017

1 Requirements

Use a current tex system! The files has been tried with texlive 2017.

2 Introduction

The luatex engine can use open type fonts but – unlike xetex which has the harfbuzz library built-in – it doesn't work with the bare binary but needs additional (lua)-code.

In LaTeX this lua-code is provided by the package `luaotfload`. In newer LaTeX-versions the lua-code of `luaotfload` is loaded by the format. So you don't need to load it manually. Even a minimal hello-world-example should show in the log-file

```
Lua module: luaotfload-main 2017/01/29 2.80001 OpenType layout system.
```

`luaotfload` consist of two parts:

- The core of the package is the *fontloader* – the current is `fontloader-2017-02-11.lua`. The fontloader has been created from a subset of the fontloader code of the ConTeXt format.
- Around this fontloader `luaotfload` contains a number of lua-file needed for the creation of the font name database, the font cache, and some adaptions of the fontloader to the needs of LaTeX.

`luaotfload` has made quite an effort to separate the two parts and since some time it is possible to replace the core fontloader with another (newer) version with the help of a configuration file. This makes it possible e. g. to test new features which have been added to ConTeXt.

---

*fischer@troubleshooting-tex.de

This package provides a number of such alternative fontloaders.

Please read the complete documentation before trying to use the fontloaders – at least read the section 6 about incompabilities.

3 Installation

The package contains

- a number of `fontloader-XXX.lua` files. This files should be in a texmf-tree in `../tex/luatex/fontloader-luaotfload` – probably your tex system installed them there already.
- all the other files are either examples or documentation and should go in the `doc`-folder.

4 Using the fontloader(s)

To use one of the alternative fontloader you must create a `luaotfload.conf` file. This file should be in the normal search path, e.g. in the folder of your document or in some texmf-tree, there at best in the branch `tex/luatex`.

Such a `luaotfload.conf` looks e. g. like this:

```
[run]
;fontloader = reference;
;fontloader = default;
;fontloader = context;
;fontloader = fontloader-reference-2017-07-28.lua;
fontloader = fontloader-reference-2017-08-18.lua;
;fontloader = fontloader-2017-02-11-stix.lua;
```

- The fontloader section starts with [run].
- The semicolon starts a comment. So in the example `fontloader-reference-2017-07-28.lua` is the active fontloader. If you comment all fontloaders `luaotfload` will use its default loader.
- The shortcuts `default` loads the default fontloader of `luaotfload` and `reference` its reference version (see below for an explanation about the reference fontloader).
- The last active entry wins!
- It is possible to use absolute pathes, but how exactly is rather system dependant, so you will have to try.

5 General description of the fontloaders

Fontloaders come in two flavors: I will call them *reference type* and *luaotfload type*.

The *reference type* is a copy from a `luatex-fonts-merged.lua` from a context installation. This file is – as the name indicates – a merge of the generic part of the context fontloader files. It is normally present in a context minimals installation, or can be created by running in the folder which contains `luatex-fonts.lua` the command:

```
mtxrun --script package --merge ./luatex-fonts.lua
```

The *luaotfload type* is created by a script call `mkimport` which you can find on the github site of `luaotfload`. It also uses the context files but ignores some that are present in the lualibs packages and it also does some other adaptions.

The original `luaotfload` already contains both types of fontloaders, the reference type is `fontloader-reference.lua` – it can be chosen in the configuration file with the keyword `reference` – and the luaotfload type is `fontloader-2017-02-11-lua`.

I have no idea about the pros and cons of both types. On the whole I would have preferred to build luaotfload type fontloaders as they are nearer to the current default fontloader. But sadly it didn't work: I could adapt the pathes in `mkimport` to my system and build a fontloader but it wasn't usable. The *reference type* are perhaps slower and there is the possibility that they contain code which could clash with some lua libraries loaded by other LaTeXpackages. But until now they seemed to work.

For both types there is the danger that fontloaders created with files from a newer context version clash with the older wrapper code of luaotfload – and actually this already happened, see the following section.

6 Incompabilities with the `luaotfload` "wrapper" files

As mentioned in the previous section it is always possible that newer versions of the core fontloader are no longer compatible to the wrapper files from `luaotfload`. If you try out `fontloader-reference-2017-07-28.lua` you will get a lua error:

```
luaotfload | load : FATAL ERROR
luaotfload | load : Failed to load module "luaotfload-letterspace.lua".
luaotfload | load : Error message:
luaotfload | load : "...mf-dist/tex/luatex/luaotfload/luaotfload-letterspace.lua:73:
                    attempt to index local 'nodepool' (a nil value)".
```

The reason is that `luaotfload-letterspace.lua` tries to access a table value that no longer exists. Imho it is not possible to solve this problem without changing the original file. This is absolutly not to my liking but as I don't see another way I offer a patched version. If you want to use the newer fontloader versions you will have to do this:

In the documentation folder of this package there is a file `luaotfload-letterspace.luax`. Change the extension to `lua` and copy the file to your document folder or in a *local* texmf tree to `tex/luatex/luaotfload` – in the second case don't forget to update your file name database (FNDB): in miktex with `initexmf -u` and in texlive with `mktexlsr`.

Try out a small lualatex document. If the log-file tells you

```
!!!!! UF: Using CHANGED luaotfload-letterspace 2017-07-14 !!!!!
```

then the new `lua`-file is used. **Don't forget that this patched file is in your system! When luaotfload is updated you should remove it.**

7 The list of new fontloaders

Currently there are only three fontloaders. But I plan to add more if some changes are made to the context files.

Fontloaders with * before their name need the adapted `luaotfload-letterspace.lua` as described in the last section.

`fontloader-2017-02-11-stix.lua` This fontloader is the original luaotfload-type fontloader where one section – from `font-oto.lua` – has been replaced to get around a bug in the stix fonts with wrong parenthese sizes. See `test-fontloader-2017-02-11-stix.tex`.

`*fontloader-reference-2017-07-28.lua` This is a reference-type fontloader which has been created on 2017-07-28. It adds the capability to add ligatures with “word boundaries”. See `test-fontloader-2017-07-28.tex`.

`*fontloader-reference-2017-08-18.lua` This is a reference-type fontloader which I added when I tried out to color the cow fonts: See https://tex.stackexchange.com/a/387069/2388 and `test-fontloader-2017-08-18.tex`.

8 Using newer luatex versions

As mentioned in the tex.stackexchange answer about the colored cow fonts, the solution needs a newer luatex. While `fontloader-reference-2017-08-18.lua` in itself seems to work fine with the luatex of TeXlive 2017, trying to use the `colr` font feature, leads to the error

```
! error:  (vf command): unknown packet command
```

So here a description how I did setup my windows system to allow tests with a newer luatex without disturbing my standard system:

1. I created a folder `.../texlive/2017/bin-dev` beside the standard `bin`-folder.
2. In this `bin-dev`-folder I copied the whole `win32` from the `bin`-folder.
3. I got `luatex-dev-w32.tar.xz` from `http://w32tex.org/` and copied the included `luatex.dll` and `lua52.dll` to the `bin-dev/win32`-folder.
4. In the `bin-dev/win32`-folder I created a `texmf.cnf`-file with the line

   ```
   TEXMFSYSVAR = $TEXMFROOT/texmf-var-dev
   ```

5. Then I changed the windows `PATH`-variable and added the path to `.../bin-dev/win32;` at the begin.
6. As a last step I called on a command `fmtutil-sys --all`. This creates the formats in a new `texmf-var-dev` tree beside the standard `texmf-var`.

With this installation it is possible to switch between the normal texlive and the one with the development luatex simply by changing the windows PATH variable.